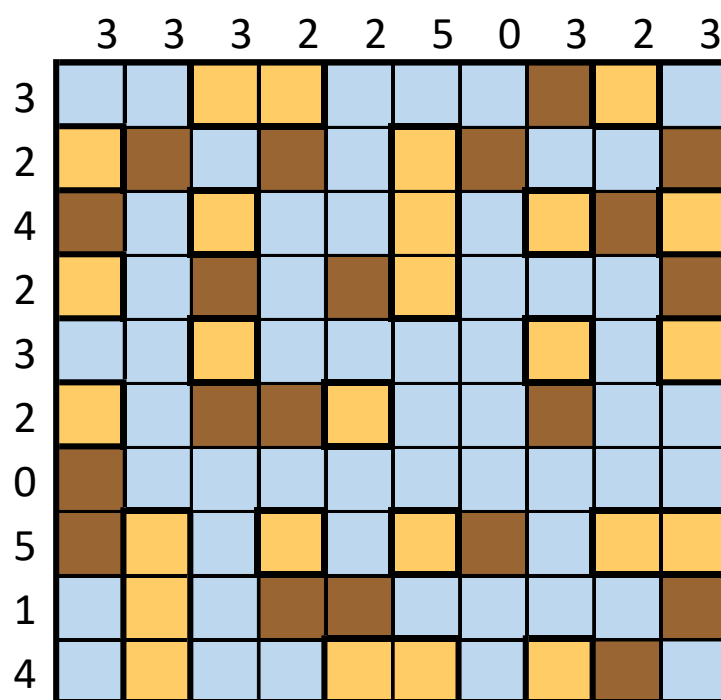


# Programação Orientada por Objetos

## Boats & Docks



Ano Letivo: 2020/2021

Projeto de Época Normal e de Recurso – 3ª Fase

# Índice

1.	Introdução.....	2
2.	Objetivos .....	3
3.	Funcionalidades e requisitos de implementação .....	3
3.1.	Interface gráfica .....	3
3.2.	Requisitos da Implementação.....	4
4.	Considerações.....	5
4.1.	Fases de desenvolvimento e entrega .....	5
4.2.	Implementação e codificação .....	5
4.3.	Constituição de grupos .....	5
5.	Entrega do projeto .....	6
6.	Regras e Critérios de Avaliação do Projeto .....	6
6.1.	Regras de Avaliação .....	6
6.2.	Critérios de Avaliação .....	7
7.	Resumo das Datas Importantes .....	8
7.1.	Entrega da 3ª fase .....	8
7.2.	Discussões de projeto .....	8

# 1. Introdução

O Projeto final pretende que o aluno sistematize os conhecimentos adquiridos de maneira a estruturar corretamente o código desenvolvido tornando-o eficiente, robusto, reutilizável e modular, para assim poder construir uma solução aplicacional de acordo com o paradigma orientado por objetos, utilizando a linguagem de programação java. O objetivo passa pelo desenvolvimento de uma versão de um jogo de tabuleiro chamado **Boats & Docks**.

O jogo **Boats & Docks** é um jogo tradicional, jogado individualmente, cuja lógica de jogo consiste numa mistura de regras entre o jogo **Minesweeper** e o **Sudoku**, sendo o principal objetivo do jogo atracar todos os barcos nos respetivos portos, seguindo um conjunto de regras. Podem consultar uma versão jogável de um jogo ([Android](#), [IOS](#)) cuja dinâmica é semelhante. Contudo, as regras do jogo a desenvolver neste projeto divergem em diversos pontos.

O projeto será desenvolvido em três fases distintas. A primeira fase consiste na análise e desenho da aplicação, resultante da implementação das classes de domínio. A segunda fase tem foco na implementação das funcionalidades e lógica de jogo, em ambiente de consola. Por último, a terceira fase será dedicada à criação da interface gráfica com o utilizador.

Tenha particular atenção ao uso do paradigma de POO na modelação das classes, i.e., ao correto uso dos conceitos de encapsulamento, herança, classes abstratas, polimorfismo, interfaces, maximização da coesão (responsabilidade única), minimização do acoplamento, desenho orientado por responsabilidades, etc. Uma modelação bem pensada facilitará todo o processo de desenvolvimento e manutenção.

O presente documento versa apenas a terceira fase de implementação do projeto. O desenvolvimento é iterativo, pressupondo o melhoramento contínuo de componentes implementadas em fases anteriores.

## 2. Objetivos

Na terceira fase do projeto o objetivo passa por desenvolver a interface gráfica, em JavaFx, que permita ao utilizador interagir com o sistema implementado nas fases anteriores.

A interface gráfica **deve incluir todos os requisitos desenvolvidos na primeira e segunda fase**, mais os indicados no ponto 3.

## 3. Funcionalidades e requisitos de implementação

Descrevem-se, nesta secção, um conjunto de funcionalidades, consideradas como requisitos funcionais, com os quais o projeto implementado deverá cumprir.

### 3.1. Interface gráfica

Deverá ser implementada uma interface gráfica, em JavaFX, que permita ao jogador efetuar todas as opções relacionadas com o jogo. Nas escolhas, referentes à implementação da interface gráfica para o utilizador, deve ser privilegiada a facilidade de navegação e a facilidade de utilização do sistema - usabilidade.

Sempre que possível, a interação com o utilizador deve contemplar mecanismos que facilitem a realização de tarefas de forma intuitiva, tais como a utilização de técnicas de seleção, “arrastamento” (*drag and drop*) e similares. O feedback visual, é importante em contextos de validação (como no caso de validação do tabuleiro de jogo, e outros), de forma a facilitar a interpretação por parte do utilizador. Devem igualmente ser adotadas boas praticas de programação defensiva, de forma a restringir os erros possíveis causados pelo utilizador.

Devem ser considerados, a criação de todos os elementos gráficos necessários, que correspondam aos requisitos funcionais implementados na primeira e segunda fase do projeto. Relativamente ao jogo, alguns dos componentes obrigatórios são:

- Nome do jogador
- Última pontuação conhecida (para o respetivo nível)
- Duração
- Grau de dificuldade
- Tabuleiro de jogo

- Botão de validação do tabuleiro (omisso até se cumprir o requisito de todos os barcos estarem colocados)
- Menu de ações (tais como: cancelar, salvar, pausar ou outras)

Para todos os restantes módulos do projeto, devem ser considerados os componentes necessários, mediante os critérios já apresentados. Qualquer aspeto omissso no presente documento, deverá de ficar ao critério do programador.

Serão **premiadas** todas as implementações com elevado grau de criatividade, cujas implementações contemplem requisitos relevantes e não especificados nos requisitos funcionais do projeto.

### 3.2. Requisitos da Implementação

Para a implementação do desenrolar do jogo deverão **obrigatoriamente ser utilizadas**:

1. Coleções observáveis
2. Propriedades (*Properties*), sempre que seja pertinente
3. Eventos para executar as alterações no ecrã
4. Efeitos ou animações (por exemplo, nas validações do tabuleiro de jogo, quer em contexto de jogo ou em situação de criação de um nível)

Em relação às fases anteriores, deverão também ser adaptados os seguintes **requisitos funcionais**:

1. Deverá ser possível definir qual a pasta, de onde deverão ser carregados os níveis de jogo
2. Deverá ser possível escolher o nome do ficheiro quando se grava o jogo ou o nível criado
3. Deverá ser possível ao jogador reiniciar todas as suas pontuações pessoais

## 4. Considerações

### 4.1. Fases de desenvolvimento e entrega

O projeto está dividido em três fases, com a cotação distribuída da seguinte forma:

- 1ª Fase: análise e design, implementação das classes de domínio (30% da avaliação final);
- 2ª Fase: lógica do problema, implementação em ambiente de consola (40% da avaliação final);
- 3ª Fase: implementação em ambiente gráfico (30% da avaliação final).

É possível proceder a melhorias contínuas do projeto submetido, em determinada fase, na entrega seguinte. No entanto, todas as melhorias deverão ser identificadas no ficheiro “README”, que acompanha o projeto.

Conforme referido na ficha da disciplina, poderá alternativamente entregar o projeto numa só data, na época de recurso (sem possibilidade de incorporar componente de avaliação contínua), contemplando as três fases do projeto.

### 4.2. Implementação e codificação

O programa deve ser desenvolvido utilizando a linguagem Java, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos.

Em relação às regras de codificação, siga as convenções adotadas normalmente para a linguagem Java:

- A notação *camelCase* para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação *PascalCase* para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize o símbolo ‘\_’ nos identificadores (exceto nas constantes), nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

### 4.3. Constituição de grupos

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo excecionalmente ser elaborado de forma individual, em situações devidamente justificadas e aprovadas. Não serão permitidos, em nenhum caso, grupos com mais do que dois alunos.

Os grupos dos alunos já se encontram determinados através da metodologia de *pair programming* que está a ser utilizada nas aulas laboratoriais. Caso existam alunos que não tenham grupo formado, deverão contactar o respetivo docente de laboratório para regularizar a situação.

## 5. Entrega do projeto

O projeto é composto por três fases distintas de implementação. A terceira e última fase, é correspondente à implementação em ambiente gráfico (descrita no presente documento), e têm de ser submetida **até às 23 horas do dia 25 de junho de 2021**.

O projeto deverá ser entregue até à data-limite especificada por **via exclusivamente eletrónica utilizando a plataforma GitHub**. Todos os ficheiros que compõem o projeto deverão estar submetidos no respetivo repositório, até à data estipulada. Em caso de dificuldades no acesso à plataforma GitHub, o envio dos ficheiros poderá ser feito por correio eletrónico para o respetivo docente de laboratório, dentro do prazo acima indicado.

**Não serão aceites quaisquer projetos entregues fora do prazo!**

Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- A **documentação do programa em JavaDoc** (não converta o documento gerado automaticamente em HTML para DOC!).
- O **manual de utilizador** da aplicação, onde deverá constar toda a informação relativa ao funcionamento e utilização do sistema.
- O **código fonte** do programa na forma de projeto em *NetBeans* ou *IntelliJ IDEA*, com um *main* de testes a funcionar (que instancie todos os objetos necessários à simulação) e com todas as funcionalidades implementadas.

## 6. Regras e Critérios de Avaliação do Projeto

### 6.1. Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- **Não serão aceites quaisquer projetos entregues fora do prazo!**

- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da Programação Orientada por Objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento da linguagem Java.
- Será valorizada a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero na fase respetiva. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respetivo professor de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem realmente cópias, serão ambos anulados, quer o projeto original quer o plágio.
- As avaliações da terceira fase do projeto serão realizadas na semana seguinte à sua submissão.

## 6.2. Critérios de Avaliação

A terceira fase do projeto terá foco na implementação da interface gráfica em JavaFx:

<b>Funcionalidades</b>	<b>50%</b>
<b>Implementação</b>	<b>30%</b>
Estrutura de classes	15%
Conhecimento e boa utilização da linguagem	5%
Bom estilo (nomes, comentários, indentação)	5%
Definição de testes unitários	5%
<b>Documentação</b>	<b>10%</b>
JavaDOC	5%
Manual de utilizador	5%



<b>Avaliação qualitativa</b>	<b>20%</b>
Qualidade geral	10%
Criatividade	10%

## 7. Resumo das Datas Importantes

### 7.1. Entrega da 3ª fase

A entrega da 3ª fase do projeto será até às 23 horas de sexta-feira, dia 25 de junho de 2021.

### 7.2. Discussões de projeto

As avaliações do projeto serão realizadas, se possível, entre os dias 28 de junho e 02 de julho de 2021.